

GENERADOR DE CONTRASEÑAS SEGURAS EN PYTHON

Josué Paucar



Propósito del Proyecto

Desarrollar una herramienta para generar contraseñas seguras que protejan datos personales y eviten accesos no autorizados.

Importancia de la Seguridad

Las contraseñas seguras previenen el robo de identidad y accesos indebidos, protegiendo la confidencialidad de los datos.

Objetivos del Programa

- Generación Automática: Crear contraseñas con letras, números y símbolos.
- Facilidad de Uso: Interfaz simple y eficiente.
- Validación de Seguridad: Verificaciones de seguridad integradas.
- Adaptabilidad: Permite ajustar la longitud de la contraseña.

DESCRIPCIÓN DEL CÓDIGO

Importación de Librerías y Definición de Variables

- Importación de librerías necesarias: `random` y `string`
- Definición de variables clave: `bandera1` y `longitud`

```
1  # Importación de Librerías
2  import random
3  import string
4
5  # Definición de Variables
6  bandera1 = True
7  longitud = int
8
```

SOLICITUD DE LONGITUD DE CONTRASEÑA

Solicitud y Validación de la Longitud de la Contraseña

- Implementación del bucle `while` para validar la longitud
- Uso de `try` y `except` para manejar excepciones

```
13  while bandera1 == True: # Se utiliza una bandera para cuando el
    ingreso del dato sea correcto salga del bucle y continúe con el
    resto del código
14      try: # Se utiliza try y except para manejar excepciones en
    caso de que el usuario pueda introducir un dato no solicitado
15          dato = input("Ingrese la longitud deseada de la
    contraseña a generar, en valor numérico entero del 8 -
    30: ")
16          longitud = int(dato) # En caso de no poder convertir
    este dato a entero, se manejará con el except
17          if 8 <= longitud <= 30: # Se verifica que el ingreso
    del entero sea entre los valores de 8 a 30
18              bandera1 = False # En el caso de que el dato
    ingresado sea correcto, la bandera se apaga y esto
    permite salir del bucle
19      else:
20          print("Por favor ingrese un número entre 8 y 30.")
21      except ValueError:
22          print("¡No se ha ingresado un carácter numérico entero!")
```

CONJUNTO DE CARACTERES

Definición del Conjunto de Caracteres

- Uso de `string.ascii_letters`, `string.digits` y `string.punctuation`
- Importancia de incluir letras, dígitos y símbolos

```
23
24 # Conjunto de caracteres disponibles para generar la contraseña
25 caracteres = string.ascii_letters + string.digits + string.
    punctuation # Conjunto de caracteres (letras, dígitos y
                símbolos) que se utilizarán
26
```


GENERACIÓN DE LA CONTRASEÑA

Generación de la Contraseña

- Se genera la contraseña utilizando una lista de compresión
- Hay un proceso de verificación de la contraseña para asegurar la inclusión de letras, dígitos y símbolos

```
27 # Bucle para generar y verificar la contraseña
28 while True:
29     # Uso de la lista de compresión para crear la contraseña
30     contraseña = ''.join(random.choice(caracteres) for i in range(longitud)) # Generación de la contraseña
31
32     # Verificación de que la contraseña tenga al menos una letra, un dígito y un símbolo
33     tiene_letra = False
34     tiene_digito = False
35     tiene_simbolo = False
36
37     for c in contraseña:
38         if c.isalpha(): # Verificar si el carácter es una letra
39             tiene_letra = True
40         if c.isdigit(): # Verificar si el carácter es un dígito
41             tiene_digito = True
42         if c in string.punctuation: # Verificar si el carácter es un símbolo
43             tiene_simbolo = True
44
45     # Si la contraseña cumple con los 3 requisitos, salir del bucle
46     if tiene_letra and tiene_digito and tiene_simbolo:
47         break # La contraseña es válida, salimos del bucle
48
49     else:
50         print("La contraseña generada no cumple con los requisitos de tener letras, números y símbolos. Generando una nueva...")
51
```

VERIFICACIÓN DE LA CONTRASEÑA

Verificación de la Contraseña Generada

- Bucle `for` para verificar la contraseña generada
- Criterios de verificación: al menor una letra, un dígito y un símbolo

```
37     for c in contraseña:
38         if c.isalpha(): # Verificar si el carácter es una letra
39             tiene_letra = True
40         if c.isdigit(): # Verificar si el carácter es un dígito
41             tiene_digito = True
42         if c in string.punctuation: # Verificar si el carácter es un símbolo
43             tiene_simbolo = True
44
45     # Si la contraseña cumple con los 3 requisitos, salir del bucle
46     if tiene_letra and tiene_digito and tiene_simbolo:
47         break # La contraseña es válida, salimos del bucle
48
49     else:
50         print("La contraseña generada no cumple con los requisitos de tener letras, números y símbolos. Generando una nueva...")
51
```

FIN DEL PROGRAMA

Resultado del Programa

- Presentación de la contraseña generada al usuario.
- Finalización del programa

```
51
52  # Impresión de La Contraseña Generada
53  print(f"Tu contraseña segura es: {contraseña}")
54
55  # Fin del Programa
56  print("-----Gracias por usar el Generador de Contraseñas Seguras-----")
57  |
```




GRACIAS